International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 7 Issue 3, May-June 2023 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Re-Engineering Traditional Learning Model with Outcome-Based Learning Curve using Embedded Training Laboratory

Ezenwa Opara; Mbonu, Ekene S.; Udemezue, Obinna E.;
Okereke, Chukwunenye G.; Uzoeto, Anurika C.

Department of Mechatronics Engineering, Federal University of Technology, Owerri, Nigeria

ABSTRACT

This paper presents the design and implementation of an indigenous outcome-based 8051 microcontroller training kit. Most of the existing microcontroller training kits available do not have local content, and also have problems with maintenance and repair when they develop faults. Users usually dump these kits when they develop faults due to a lack of trained technical maintenance personnel. This project used components that are locally available to implement an outcome-based 8051 microcontroller training kit that is modular and easy to use. In the methodology, the bottom-top approach was used to design the individual modules of the kit before fabricating them on a printed circuit board (PCB). The user manual was then developed and experiments in it were tested using the developed training kit. The result of the work is an outcome-based 8051 microcontroller training kit that meets the National Universities Commission (NUC) and Council for the Regulation of Engineering in Nigeria (COREN) standards. It has various modules for the user to practice several 8051 microcontroller programming experiments. The work can be deployed in university laboratories for microcontrollers and embedded systems training. Specifically, there are courses in Mechatronics engineering, FUTO, that can be taught using this kit.

KEYWORDS: *Microcontroller, 8051, motherboard, circuit*

How to cite this paper: Ezenwa Opara | Mbonu, Ekene S. | Udemezue, Obinna E. | Okereke, Chukwunenye G. | Uzoeto, Anurika C. "Re-Engineering Traditional Learning Model with Outcome-Based Learning Curve using Embedded Training Laboratory" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-7 | Issue-3, June 2023, pp.163-173, URL: www.ijtsrd.com/papers/ijtsrd55127.pdf

IJTSRD55127

I. INTRODUCTION

In the art of programming on a desktop and learning how to program microcontroller systems, you might be surprised to find that programming microcontrollers are quite different from what you are used to, and what you have learned before. For example:

- Most microcontroller systems do not have a graphic user interface (GUI).
- The microcontroller system might not contain any operating system (typically this is called bare metal). Or, in some instances, a lightweight RTOS is used, which only manages task scheduling and inter-task communication. Unlike desktop environments, many of these operating systems do not provide other system Application Programming Interfaces (APIs) for data communication and peripheral control.
- In desktop environments, the applications access peripheral functions via APIs or device drivers provided in the OS. Whereas in microcontroller applications, it is not unusual to access the peripheral registers directly. However, most Cortex-M microcontroller vendors also provide device driver libraries to make it easier for software developers to create their applications.
- Memory size and power consumption are constraining factors in many microcontroller systems. In contrast, the amount of memory and processing power in a desktop environment is significantly greater.
- In desktop environments, the use of assembly language is quite rare, and most application developers use a wide range of high-level programming languages, including Java/JavaScript, C#, and Python. Today, most

microcontroller projects are still based on C and C++. In some instances, a small portion of the software could be written in assembly language [1].

In the literature discussing microprocessors, we often see the term embedded system. Microprocessors and microcontrollers are widely used in embedded system products. An embedded product uses a microprocessor (or microcontroller) to do one task and one task only. A printer is an example of an embedded system since the processor inside it performs only one task; namely, getting the data and printing it. Contrast this with a Pentium-based PC (or any x86 IBM-compatible PC). A PC can be used for any number of applications such as a word processor, print server, bank teller terminal, video game player, network server, or internet terminal. Software for a variety of applications can be loaded and run. Of course, the reason a PC can perform myriad tasks is that it has RAM memory and an operating system that loads the application software into RAM and lets the CPU run it. In an embedded system, there is only one application software that is typically burned into ROM. An x86 PC contains or is connected to various embedded products such as the keyboard, printer, modem, disk controller, sound card, CD-ROM driver, mouse, and so on. Each one of these peripherals has a microcontroller inside it that performs only one task. For example, inside every mouse, there is a microcontroller that performs the task of finding the mouse position and sending it to the Pc [2].

II. PROBLEM STATEMENT

Training kits and lab equipment shipped and sold in Nigeria have significant low maintenance rate due to lack of local contents and unavailability of trained maintenance personnel associated with such kits and equipment. Thus, users are forced to either ship the product abroad when they develop faults or dump them and get new ones. In addition, these products are shipped with ambiguous user guide/manual which do not conform with the COREN or NUC standards.

Thus, there is need to develop and produce training kits whose components should not only be sourced locally, but the maintenance personnel should be readily available. This project is an attempt to solve the identified problems with respect to 8051 microcontroller training kit.

III. AIMS AND OBJECTIVES

The aim of this project is to design and implement an outcome based 8051 microcontroller kit with a well-documented manual. In order to achieve the above aim, the work shall be geared towards the following objectives:

1. To interface:
   - LCD to 8051 microcontrollers.
   - LEDs to 8051 microcontrollers.
   - Seven segment display to 8051 microcontrollers.
   - Keypad to 8051 microcontrollers.
   - Motors to 8051 microcontrollers.
   - AC load to 8051 microcontrollers.
2. To integrate the various interfaces of the controller kit into a single design.
3. To produce the PCB for the 8051 microcontroller kit.
4. To produce an outcome-based manual for common experiments on 8051 microcontrollers.

IV. REVIEW OF LITERATURE

The authors H. Apaydin and N. Fusun [3] presented a paper where the design of the 16F877 microcontroller experiment set was realized. They have described and performed a novel learning environment design to comprehend the response of electric circuits. The modules in the experiment set are similar to those in the ISIS program. Because the modules are similar to the circuit diagrams of the simulations, students will not be strangers when practicing in the real environment. At the same time, the students will be able to reinforce what they have learned by using the experiment set outside the school. All the same, their system failed to expose the students to the programming part of microcontroller but only focused on the schematic connections and simulation.

In [4] miniaturization is the main contribution of Edukit which is module based and used 8051 microcontroller. An ice-cream size box contains the whole Edukit which proves that the Edukit is simple, portable, low cost, consuming less power.

Moreover, the Edukit is a module based which in turn makes it a low-cost and suitable educational tool for any discipline of science and engineering. This is a brilliant one with its modularity and multi-module but lacks the ability for a user(s) to follow a well-documented laboratory manual to perform the various experiments associated the modules.

The product [5] Unity Board is a platform that brings both Professionals and Students (Industry and Academia) together; made-in-Nigeria IoT (internet of things) fully featured Development Board & Educational Kit bringing the internet of things at your fingertips for every user. With the possibility of talking to the cloud, great opportunities are created like using smart technologies to solve unique problems and design revolutionary innovations. It is user-friend and fully compatible with Arduino and peripheral modules making it programmable with

Arduino IDE and compatible with all the example codes and libraries for Arduino. This product fails to acquaint the user(s) especially students who are new to the basics of microcontroller programming and interfacing. It is focused on IoT and would be appreciated more by professionals who are already armed with the fundamentals of microcontroller.

The main purpose of the paper [6] is designing a suitable educational kit to educate beginner with basic programming algorithm using microcontroller which involves hardware and software. This new expansion board is suitable for newcomers who want to learn microcontroller programming. Primary school children can use scratch programming to learn basic microcontroller programming. Secondary or higher-level student can use C++ languages to write the microcontroller program. The system is modular and has an accompanying laboratory manual for effective learning of the laboratory sessions associated with the board. However, the trainer board runs on Arduino which is intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments [6] and not tailored to undergraduates who wish to really learn and understand microcontroller programming basics.

In [7] the paper explains the design and development of Personal Computer Less (PC-Less) Microcontroller Training Kit. It was developed based on Lattepanda processor and Arduino microcontroller as target. The training kit is equipped with advanced input-output interfaces that adopted the concept of low cost and low power system. The preliminary usability testing proved this device can be used as a tool for microcontroller programming and industrial automation training. By adopting the concept of portability, the device could be operated in the rural area where electricity and computer infrastructure are limited. Furthermore, the training kit is suitable for student of electrical engineering student from university and vocational high school. Nevertheless, the system is Arduino based and failed to expose students to the basics of microcontroller programing especially the control of I/O ports.

The of authors of the literatures reviewed have done commendable works based on their focus. However, there still exists a knowledge gap of how this wonderful equipment would be used. Many of the researchers focused on implementation without considering the users of the trainer board. A perfect example is the work of [4] which produced a multi-module trainer kit but lacked a user manual.

This research focuses on modularity and production of self-explanatory laboratory manuals which help the user(s) to perform all the possible experiments associated with the trainer kit.

V. MATERIALS AND METHOD

A. Materials

The following tools were used to carry out this project. For clarity, the list has been subdivided into hardware and software tools.

- Hardware tools

Multimeter
Soldering iron
Soldering lead
PCB copper board
PCB etching solution
Driller and drilling bits
Cutter
Plier
Screw driver

- Software tools

Proteus Design Suite
M-IDE Studios for MCS-51
Top-win programmer software

B. Methods

A. Interfacing LCD to 8051 microcontroller

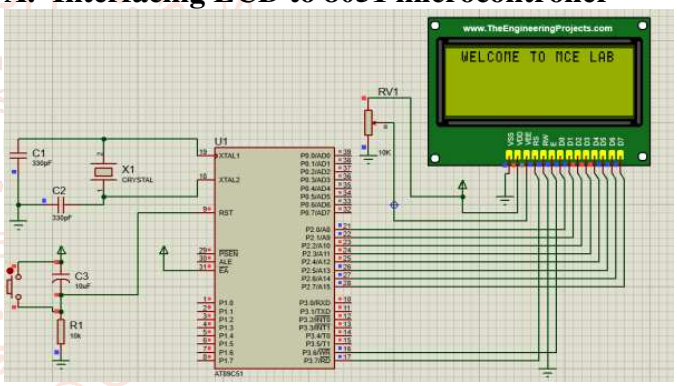

Figure 1: Interfacing LCD to 8051 microcontroller

The circuit diagram given in figure 1 shows how to interface a 20×4 LCD module with AT89c51 microcontroller. Capacitor C3, resistor R1 and push button switch form the reset circuitry. Ceramic capacitors C1, C2 (33pF each) and crystal X1 is related to the clock circuitry which produces the system clock frequency. P2.0 to P2.7 pins of the microcontroller is connected to the DB0 to DB7 pins of the LCD module respectively and through this route, data are sent to the LCD module. P3.6 and P3.7 are connected to the enable (E), register select (RS) pins of the microcontroller, while read/write (RW) pin is connected to ground (constant LOGIC 0), and through these pins the control signals are transferred to the LCD module. The potentiometer RV1 is used for adjusting the contrast of the LCD. Detailed procedure for programing of LCD can be found in Appendix A.

Table 1 below shows a summary of the LCD pin description, a detailed explanation of LCD pin description, LCD initialization is contained in appendix A.

Table 1: Pin Description of 20X4 LCD

Pin No.	Name	Function
1	VSS	This pin must be connected to the ground
2	VCC	Positive supply voltage pin (5V DC)
3	VEE	Contrast adjustment
4	RS	Register selection
5	R/W	Read or write
6	E	Enable
7	DB0	Data
8	DB1	Data
9	DB2	Data
10	DB3	Data
11	DB4	Data
12	DB5	Data
13	DB6	Data
14	DB7	Data
15	LED+	Back light LED+

- **20×4 LCD module commands**

20×4 LCD module has a set of preset command instructions. Each command will make the module to do a particular task. The commonly used commands and their function are given in the table 2 below.

Table 2: Commonly used commands in LCDs

Command	Function
0F	LCD ON, Cursor ON, Cursor blinking ON
01	Clear screen
02	Return home
04	Decrement cursor
06	Increment cursor
0E	Display ON, Cursor blinking OFF
80	Force cursor to the beginning of 1stline
C0	Force cursor to the beginning of 2nd line
38	Use 2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, Cursor OFF
C1	Jump to second line, position1
OC	Display ON, Cursor OFF
C1	Jump to second line, position1
C2	Jump to second line, position2

- **Sending data to the LCD.**

The steps for sending data to the LCD module is given below. As already discussed, LCD module has pins namely RS, R/W and E. It is the logic state of these pins that make the module to determine whether a given data input is a command or data to be displayed.

- Make R/W low.
- Make RS=0 if data byte is a command and make RS=1 if the data byte is a data to be displayed.
- Place data byte on the data register.
- Pulse E from high to low.
- Repeat above steps for sending another data.

B. Interfacing LED to 8051 microcontrollers

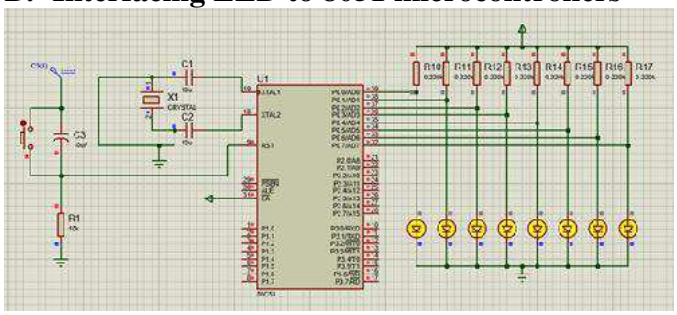

Figure 2: Interfacing LED to 8051 microcontroller

Figure 2 shows the interface of LEDs to the microcontroller. The components used for the interface are:

- AT89C51 (8051 Microcontroller)
- 8 LEDs
- 8 Resistors – 1KΩ
- Crystal oscillator – 11.0592MHz
- 2 Capacitors – 33pF
- 2 Resistors – 10KΩ
- 1 Capacitor – 10µF
- 1 Push Button
- 8051 Programmer
- 5V Power Supply

Circuit Design

In the circuit, LEDs are connected to the port P0 of the MCU and are pulled up through a 0.220k ohms resistor. As shown in figure 3.2 above, an external crystal oscillator is connected to the MC through pin 18 and 19. Crystal pins are connected to the ground through two capacitors of 10uF.

How to Control LEDs

LEDs are semiconductor light sources. Commonly used LEDs have biasing voltage of 1.7V and working current of 5mA-30mA. When an LED is applied with its required voltage and current, it glows with full intensity. The LED is similar to the normal PN diode but it emits energy in the form of light. The color of light depends on the band gap of the semiconductor. The LED is connected to the AT89C51 microcontroller with the help of a current limiting resistor. The value of this resistor is calculated using the following formula.

R= (V-1.7)/15mA, where V is the input voltage.

Generally, microcontroller outputs a maximum voltage of 5V. Thus, the value of resistor calculated for this is 330 Ohms. This resistor can be connected to either the cathode or the anode of the LED.

How to program the LEDs

The details of how to program the LEDs are shown in appendix A.

C. Interfacing Seven Segment Display to 8051 Microcontroller

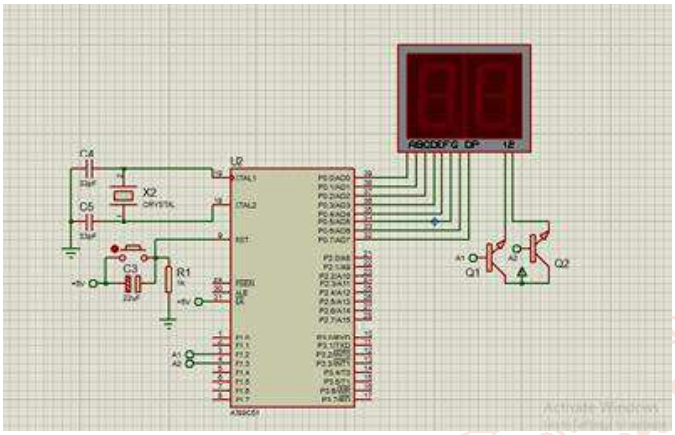

Figure 3: Interfacing seven segment

The schematic in figure 3 shows the interface of a double digit seven segment display to an 8051 microcontroller (89C51). The code for this can be found in appendix B; seven segment display source code.

Circuit Components

- AT89C51 Microcontroller
- AT89C51 Programming board
- Programming cable
- 12V DC battery or adaptor
- Common Cathode 7 segment Display
- Resistors – 10KΩ X 2, 330Ω, 1KΩ X 8
- 1KΩ X 8 Resistor Pack
- 33pF Ceramic capacitors x 2
- 11.0592 MHz crystal
- 10µF Electrolytic capacitor
- Push button
- Connecting wires

Circuit Design

In this circuit, pins a to h of the 7 segment are connected to the PORT 1 of the 89c51 MC and *com* pin is connected to the ground through the 220-ohm resistor. This resistor is used to drop the voltage. Since a common cathode seven segment is used, LOGIC 1 is sent to the segments to glow.

The operating voltage of this LED's is 2V to 3V but from controller we will get 5V so to drop the remaining voltage we have to connect a to g pins to the controller through the resistor.

Digit Drive Pattern

To display digits on 7 segment, we need to glow different segments. For instance, to display three (3) on seven segment you need to glow these segments a, b, c, d and g. Table 3.3 shows the Hex values sent from PORT1 to Display digits from 0 to 9.

Table 3: Display numbers on a seven segment display in common cathode configuration

Number	g f e d c b a
0	0 1 1 1 1 1 1
1	0 0 0 0 1 1 0
2	1 0 1 1 0 1 1
3	1 0 0 1 1 1 1
4	1 1 0 0 1 1 0
5	1 1 0 1 1 0 1
6	1 1 11 1 0 1
7	0 0 00 1 1 1
8	1 1 11 1 1 1

How to program seven segment display

A complete guide on how to operate the seven segment display is contained in appendix A.

D. Interfacing Keypad to 8051 microcontrollers

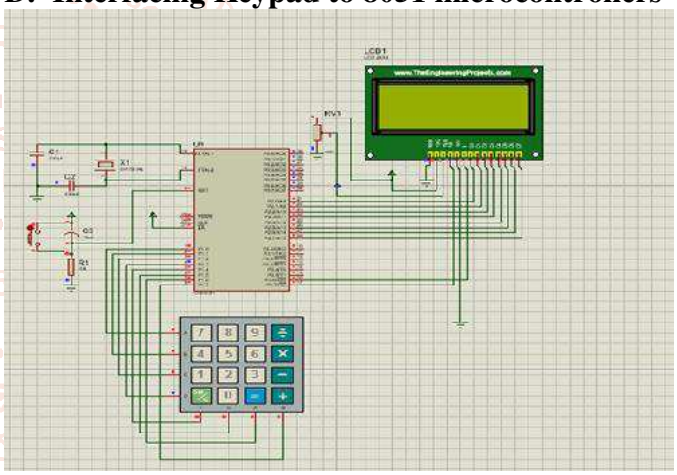

Figure 4: Proteus Simulation of Interfacing 4*4 matrix keypad with 8051

Circuit Design

In the circuit, Keypad is connected to the port P1 of the MCU and are pulled up through a 0.220k ohms resistor. As shown in figure 4 above, an external crystal oscillator is connected to the MC through pin 18 and 19. Crystal pins are connected to the ground through two capacitors of 10uF. The columns of the keypad are connected to the upper nibbles of P1 and the rows connected to the lower nibbles of P1. Also the LCD is connected is connected to P2 in other to display the results of the operation of the Keypad.

The components used for this circuit includes;

- AT89C51 (8051 Microcontroller)
- Keypad module
- Crystal oscillator – 11.0592MHz
- 2 Capacitors – 33pF

- 2 Resistors – 10KΩ
- 1 Capacitor – 10µF
- 1 Push Button
- 8051 Programmer
- LCD module
- 5V Power Supply

How to program a keypad

A well detailed procedure on how to program a keypad is contained in appendix A.

E. Interfacing Motors to 8051 microcontroller

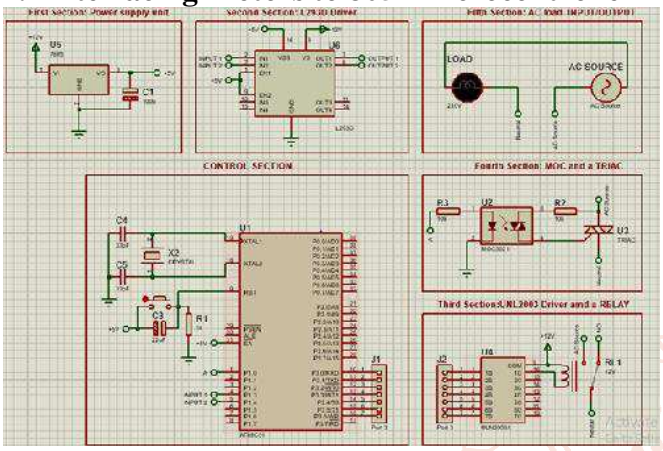

Figure 5: Driver board interfaced with the Motherboard

Figure 5 above shows the circuit diagram of the interface of motor drivers, AC load and 8051 microcontroller. Capacitor C3, resistor R1 and push button switch forms the reset circuitry. Ceramic capacitors C1, C2 (330pF each) and crystal X1 is related to the clock circuitry which produces the system clock frequency. The maximum output current of microcontroller pin is 15mA at 5V. But the power requirements of most DC motors is out of this range. Also, the back emf (electromotive force) which is produced by the motor may damage the microcontroller. Hence, the need to interface DC motors with microcontroller through drivers. L293D Driver, UNL2003 Driver and MOC 3021 were used as contained in figure 3.5. In other to achieve modularity, all the motor drivers were mounted on a separate board referred to as the driver board.

Components used

- AT89C51 (8051 Microcontroller)
- Relay
- Drivers (MOC 3021, UNL2003, L293D)
- Dc motor (DC motor, servo motor and stepper motor)
- Crystal oscillator – 11.0592MHz
- 2 Capacitors – 33pF
- 2 Resistors – 10KΩ
- 1 Capacitor – 10µF
- 1 Push Button
- 8051 Programmer
- 5V Power Supply

For clarity, a full description of the interfacing of the Dc motors, AC load with 8051 microcontroller (89c51) through the respective drivers are given below.

F. Interfacing servo motor to 8051 microcontroller through L293D Driver

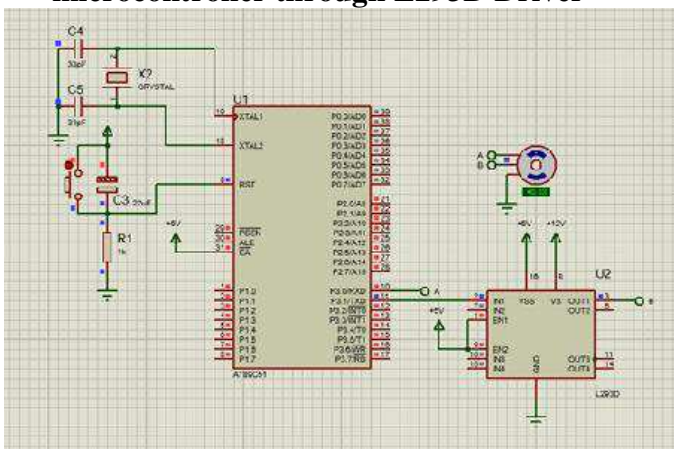

Figure 6: Interfacing servo motor with MCU

Circuit explanation

As shown in figure 6, the servo motor is connected to port 3 of the 8051 MC through the IC L293D driver. Pin 19 and 18 of the MC are used to set the clock frequency of the microcontroller while pin 9 is the reset pin. The control wire of the servo motor is connected to P3.0, through this pin the microcontroller controls the degree of rotation of the servo (as programmed). The VCC (power supply) wire of the servo is connected to OUT1 of the L293D and the last wire of the servo is grounded. Also pin 1(EN1), 9(EN2) and 16(VSS) of IC L293D is connected to a 5v power supply. A full description of how to program and operate a servo motor is contained in appendix A.

G. Interfacing DC motor to 8051 microcontroller through L293D

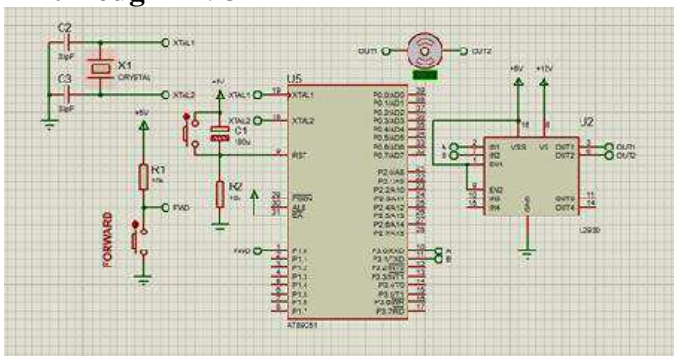

Figure 7: Interfacing DC motor with MCU

Circuit Explanation

The 8051 microcontroller by default has all the pins high. As shown in figure 7 Pin 18 and 19 were used to set the controller's clock frequency. A crystal oscillator is used parallel with two capacitors. Pin 9 is called the RESET pin, which is used to reset the microcontroller to its initial values. From the circuit

diagram, pin 9 is configured to receive a low when the push button is pressed (there was no code written for this operation, it's a hardware function by default). Port 1.0 has been configured to receive a HIGH by default, unless the FORWARD button is pressed.

From figure 7, the two input pin of L293D (i.e. IN1 and IN2) are connected to P3.0 and P3.1 of the microcontroller. this is the channel through which the microcontroller sends in high and low signals. From the assembly code written, the 8051 was programmed to always send a high as default to keep the DC motor moving in a backward direction, but when the push button is pressed the signal goes low thereby telling the 8051 to send a low signal to the driver which changes the direction of the motor to forward. The control program for this operation is contained in appendix B.

H. Interfacing Stepper motor with 8051 microcontroller through UNL2003

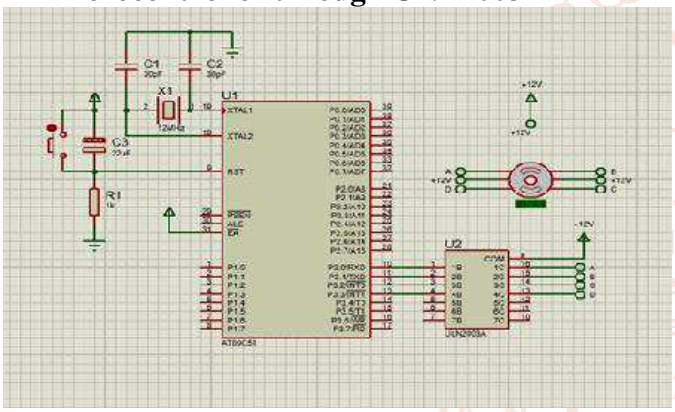

Figure 8: Interfacing stepper motor with MCU

Circuit design

To Interface stepper motor with 8051, we just need to give 0 and 1 to the four wires of stepper motor depending on which mode we want to run the stepper motor. As shown in figure 3.8, the stepper motor is connected to P3 of the microcontroller through UNL20003A. an external crystal oscillator is connected to the MC through pin 18 and 19, Crystal pins are connected to the ground through two capacitors of 10uF. The four wires of the motor A, B, C and D are connected to the output ports (1C – 4C) of the UNL2003A driver for receiving amplified signal, while the remaining two wires are connected to pin 9(COM) of the driver this serves as the power supply to the motor (12V). Four input pins (1B – 4B) of the driver is connected to the first four pins of P3 (i.e. P3.0, P3.1, P3.2 and P3.4) respectively and rest two wires should be connected to a proper 12v supply (depending on the stepper motor). Here we have used the unipolar stepper motor. We have connected four ends of the coils to the first four pins of port 2 of 8051 through the ULN2003A. ULN2003A is the array of seven NPN Darlington transistor pairs. Darlington pair is constructed by connecting two bipolar transistors to achieve high current amplification. In ULN2003A, 7 pins are input pins and 7 pins are output pins, two pins are for Vcc (power supply) and Ground. Here we are using four input and four output pins.

I. Interfacing AC Load to 8051 microcontroller

Figure 9 shows how to interface AC load to MC this is achieved through two methods;

By interfacing the AC load to the controller via a driver (UNL2003) and a relay RL1

Through a TRIAC BT138 and its driver MOC 3021

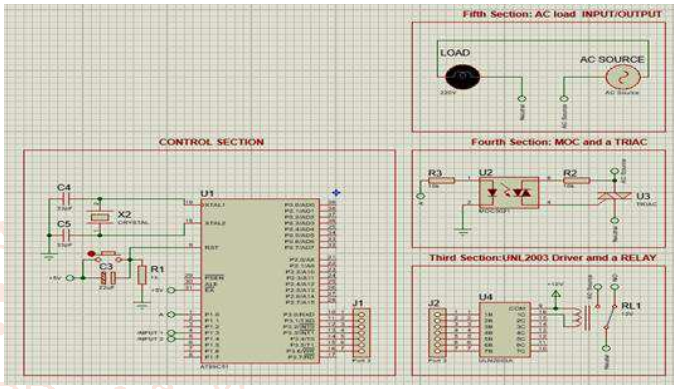

Figure 9: Interfacing AC driver with MCU

J. Integrating the Various Interfaces of the Controller into a Single Design

The overall integration of the various components of the controller kit is presented here. Figure 10 below shows the circuit diagram of the entire kit, with each section clearly labelled.

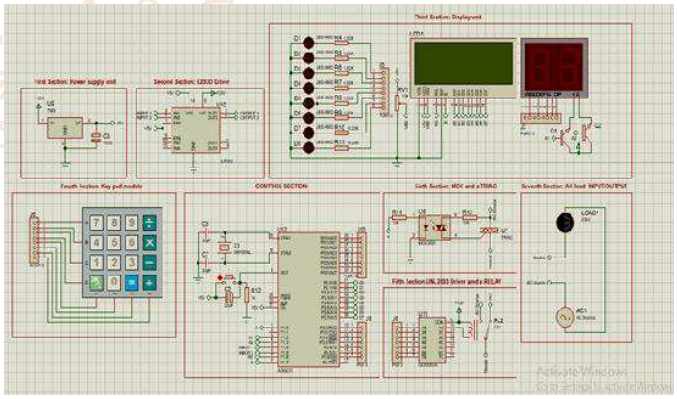

Figure 10: Overall circuit design for the microcontroller training kit

The control section

This section contains the 8051 microcontroller (89c51), it is the brainbox of the training kit. All other components of the kit are interfaced to the ports of the MCU to carry out experiments on the board. As already explained, the MCU contains external circuit like the crystal oscillator which accounts for the oscillating frequency of the MC and also the reset button which is connected to pin 9 of the MC and pulled down through a 1k resistor. Pin 1.0 is also

connected to a push button, which can be programmed to do something. The remaining ports are looped out to a connector so they be interfaced with other components. The MCU is directly mounted on the motherboard of the training kit alongside other components like LEDs, LCD and seven segment display unit.

First section (Power Supply Unit)
This is the part of the system that regulate and filters the input voltage.

Second, fifth and sixth section (UNL 2003, MOC 3021 and L293D)
These section shows the different drivers which has already been discussed earlier, these drivers are all mounted on a separate board called the daughter board or driver board. It is used for interfacing Dc motors and Ac load to the microcontroller.

Third section (Display Unit)
This section contains the output devices used in the training kit. A detailed explanation of the working principles of these devices are given in the previous sections. The output devices covered I this design include LEDs, LCDs and double digits seven segment display. These devices are mounted directly on the motherboard of the training kit alongside the MCU.

Fourth section (keypad module)
This is a separate module on its own. The keypad is the primary input device to the microcontroller, the principle of operation and programming of the keypad has been discussed in the section above.

K. Manual for common experiments on 8051 microcontrollers

To fulfil this objective, a laboratory guide containing step by step procedures on how to operate and carryout several practical on the training kit is attached to this report as appendix A. Also, a storage device containing all the firmware for different modules of the training kit has been provided.

L. System Testing

The system consists of two parts: the hardware and the software. Various tests were carried out on these parts to ensure good working of the system.

M. Hardware Testing

As earlier stated, the hardware part of the system is in modules. The various modules were first tested separately, then integrated as required and also tested to ensure compatibility.

N. Testing the Motherboard

On the motherboard, continuity test was first carried out to ensure that all the components are properly connected. The motherboard was then powered on through the power supply unit.

O. Testing the hardware modules

Continuity test was also carried out on each of the separate hardware modules and the necessary corrections were made. With the motherboard powered off, the 8051 microcontroller was programmed and mounted on the motherboard in the appropriate position. the motherboard was powered on. This was done successively for each module being tested and it was observed and results noted.

P. Software Testing

The software testing involved several processes. First the various software to write and upload the code was downloaded. M-IDE was used to write and compile the assembly code. The code for each module was written, compiled to hex file and burned to the microcontroller. In burning the code, the Topall8 software was used together with the hardware programmer connected to the PC. Several subroutines written to ensure that each module works fine. Initially, a lot of syntax errors were encountered but the code was carefully debugged over and over until everything started as expected. Some of the sample codes are shown in Appendix A.

VI. STRUCTURED ALGORITHM

Several technologies are involved in programming the microcontroller. The programming code first has to be written, compiled and burned to the microcontroller before mounting it on the motherboard. The 8051 Assembly language or C can be used to program the chip and the following are the software used in writing and compiling the code:

- Top Win Software with the programmer
- M-IDE

Algorithm 1. Programming the LCD

```
input: initialization of LCD
    MOV A,#38H          ;init. LCD 2 lines, 5x7 matrix
    ACALL COMNWRT       ;call command subroutine
    ACALL DELAY         ;give LCD some time
    MOV A,#0CH          ;display on, cursor on
    ACALL COMNWRT
    ACALL DELAY
    MOV A,#01   ;clear LCD
    ACALL COMNWRT
    ACALL DELAY
    MOV A,#06H;shift cursor right
    ACALL COMNWRT
    ACALL DELAY
    RET

        output: alphanumeric characters
                Series of alphanumeric characters and
                custom characters output.

        parameters: LCD_DATA, RS, ENABLE, DELAY
while controller sends data to LCD_DATA
    do: continuously output the bits (characters) sent to the
    output pin.
    Check the RS pin.
    Execute DELAY
    Check other I/O pins
END
```

Algorithm 2. Controlling a stepper motor

input: initialization of motor output pins

output: rotate the stepper motor

parameters: STEP_OUT_PIN, DELAY
while true
do: continuously move the motor through a step and delay
end

VII. RESULTS AND DISCUSSION

A. Result of interfacing LCD to 8051 microcontrollers

Interfacing LCD to 8051 microcontrollers board was realized and shown in figure 11 below.

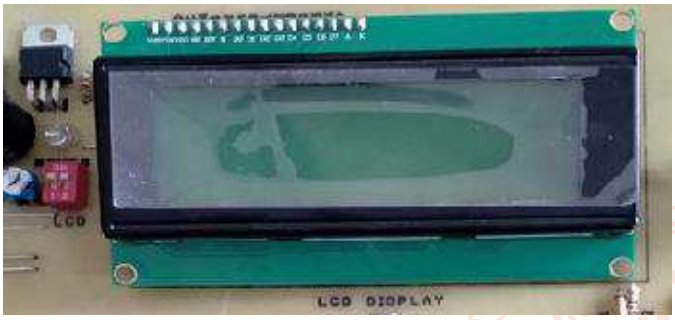

Figure 11: LCD Module on the Motherboard

B. Result of interfacing LEDs to 8051 microcontroller

Figure 12 below shows the hardware result of the LEDs connected to the 8051 microcontroller board.

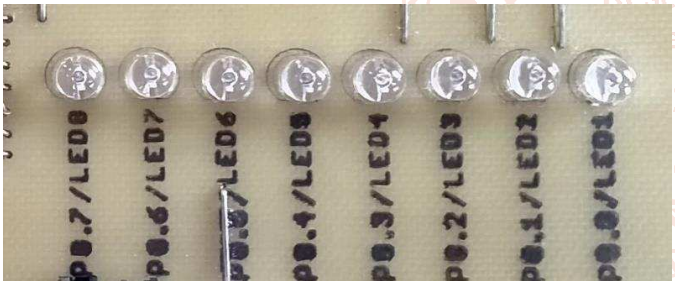

Figure 12: LED Module on the Motherboard

C. Result of interfacing seven segment display to 8051 microcontrollers

The seven segment display interfaced to the motherboard is shown in figure 13 below.

Figure 13: Seven segment display Module on the Motherboard

D. Result of interfacing Keypad to 8051 microcontrollers

Figure 14 below shows the interfacing of 4*4 Keypad to 8051 microcontrollers.

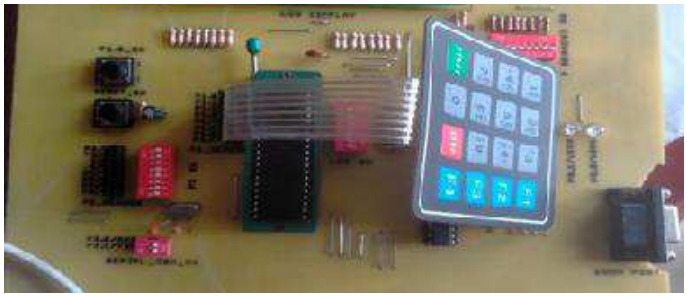

Figure 14: Interfacing Keypad to 8051 microcontrollers

E. Result of interfacing Motors to 8051 microcontrollers

Figure 15 below shows the result of interfacing DC motor to the board using the driver module.

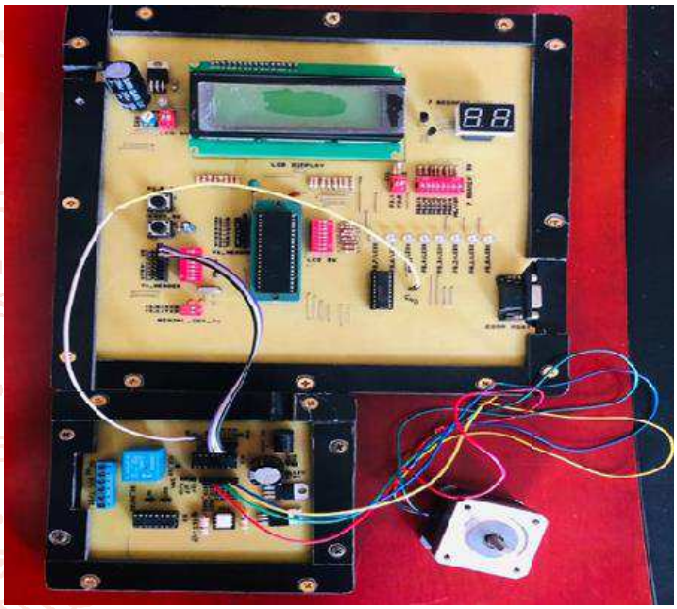

Figure 15: Interfacing DC motor to the board using the driver module

F. Result of integrating the various interfaces of the controller kit into a single design

Figures 16 and 17 below shows the integration of various modules to the motherboard.

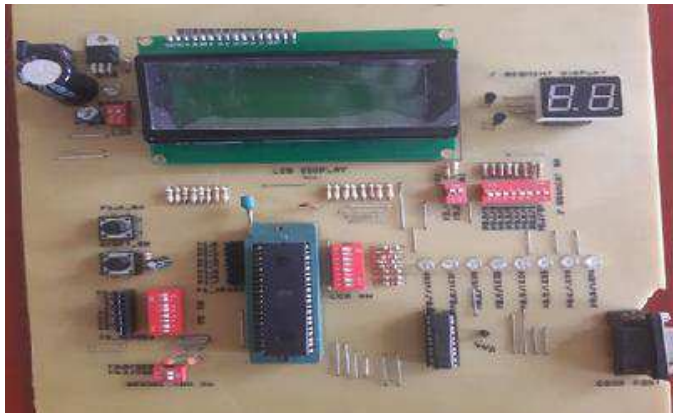

Figure 16: Integration of seven segment and LCD modules to the motherboard.

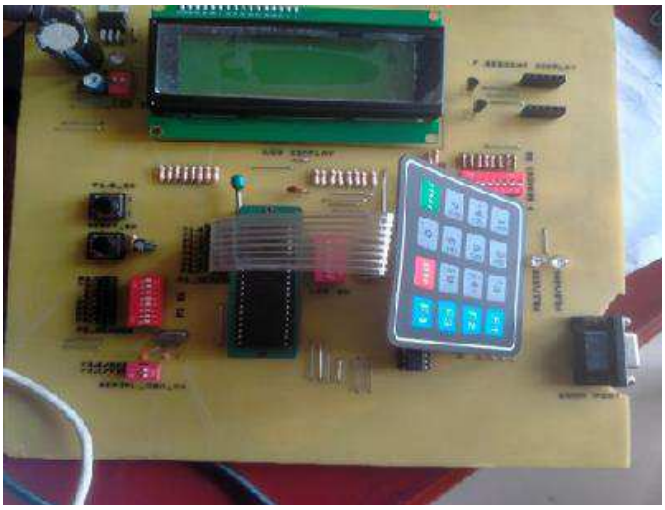

Figure 17: Integration of keypad and LCD modules to the motherboard.

G. Production of a manual for common experiments on 8051 microcontrollers

The accompanying manual for common experiments can be found in Appendix A.

VIII. DISCUSSION OF RESULTS

Following the procedure of practical 2.1 Blinking LEDs of the manual in appendix A, the code was compiled and downloaded into the microcontroller chip. The result shown in figure 18. clearly proves that both the code and experimental procedures worked very well. Other modules were also tested and were confirmed okay.

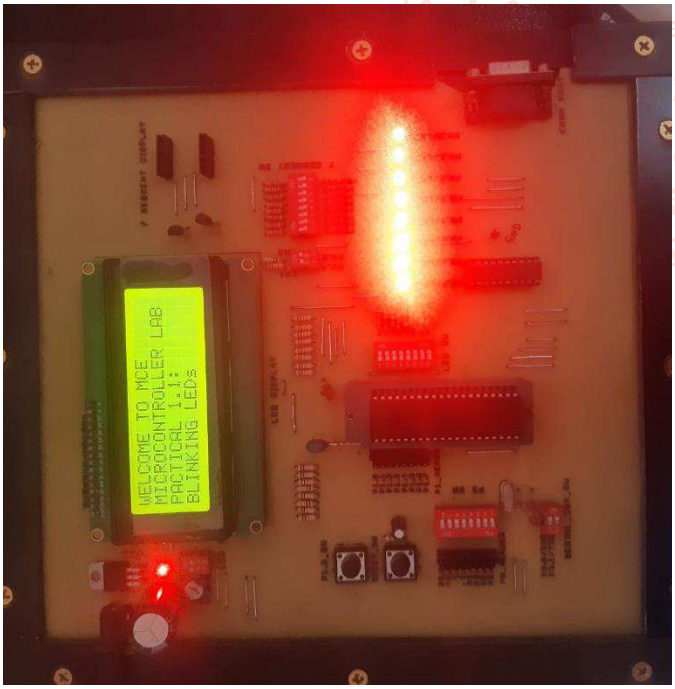

Figure 18: Output of LCD test

IX. CONTRIBUTION TO BODY OF KNOWLEDGE

This work made the following contributions to the body of knowledge.

- It will bring about a more detailed evaluation and understanding of the 8051 microcontroller which will in turn increase the interest of students in the study of microcontrollers.
- With the well-developed study and user manual, lecturers can comfortably teach microcontroller related courses properly and in details and students in turn will be well equipped practically.
- With the use of assembly language in this work, students will be more acquainted with the basics of programming language.
- With the execution of this work, the approach of building from scratch and maintenance will be imbibed in students.

X. RECOMMENDATIONS

Embedded systems are important in mechatronics engineering as many mechatronic systems have embedded systems on them. It is encouraged that students at their undergraduate level should learn and master embedded system programming. This project provides a solid foundation for starters and professional. Universities, skill acquisition and private users are encouraged to deploy this kit to their labs and be effectively utilized.

Training boards like this with recent microcontroller should be developed to enable researchers and learners practice and learn with later model microcontrollers.

XI. CONCLUSION

This work has successfully produced an outcome-based 8051 microcontroller training kit with a well detailed user manual. The result of this work is recommended for mass production.

XII. REFERENCES

[1] J. Yiu, Definitive Guide to Arm Cortex-M23 and Cortex-m33 Processors, 2021.

[2] J. G. M. a. R. D. M. Muhammad Ali Mazidi, The 8051 Microcontroller and Embedded Systems Using Assembly and C, Prentice Hall, 2006.

[3] N. F. O. S. H Apaydin, "Microcontroller Training Kit Design Compatible with Drawings of the ISIS Simulation Program," *International Journal of Education and Information Technologies,* vol. 14, pp. 22-30, April 2020.

[4] L. R. a. S. A. Liakot Ali, "Module-based Edukit for teaching and learning 8051 microcontroller programming," *2017 IEEE International Conference on Telecommunications and Photonics (ICTP),* pp. 57-61, 2017.

[5] "About Unity Board," 2020. [Online]. Available: http://unityboard.ng/about-unity-board/. [Accessed March 24th 2021].

[6] V. Murugesu, "Design an Expansion Board for Arduino Uno Microcontroller Development Board with Multiple Input and Output," *SkillsMalaysia Journal,* pp. 1-34, 2019.

[7] W. a. I. F. Y Somantri, "Personal Computer-less (PC-less) Microcontroller Training Kit," *IOP Conference Series: Materials Science and Engineering,* 2018.

[8] "Introduction to microcontroller - OpenLabPro.com," 2020. [Online]. Available: https://openlabpro.com/guide/basics-of-microcontroller/. [Accessed 18th March 2021].

[9] Ibrahim, "A new approach for teaching microcontroller courses to undergraduate students," *Elsevier,* 2014.

[10] "Evolution of MCU," 2020. [Online]. Available:https://semiengineering.com/evolution-of-the-mcu/. [Accessed 18th March 2021].

[11] "Evolution of the Microcontrollers," [Online]. Available: tesla-institute.com/index.php/microcontrollers/78-4-ways-to-save-electricity. [Accessed 18th March 2021].

[12] "Microcontrollers - 8051 Architecture," 2020. [Online]. Available: https://www.tutorialspoint.com/microprocessor/microcontrolle rs_8051_architecture.htm. [Accessed 23rd March 2021].

[13] "Basics of Microcontroller," 2020. [Online]. Available:https://www.electronicshub.org/microcontrollers-basics-structure-applications/. [Accessed 18th March 2021].

[14] "LED Interfacing," 2020. [Online]. Available: https://www.elprocus.com/led-interfacing-with-8051-microcontroller/. [Accessed 17th May 2021].

[15] "LEDsmagazine," 1st September 2004. [Online]. Available: https://www.ledsmagazine.com/leds-ssl-design/materials/article/16701292/what-is-an-led. [Accessed 17th May 2021].

[16] "Electrosome," 2020. [Online]. Available: https://electrosome.com/interfacing-lcd-with-8051-using-keil-c-at89c51/. [Accessed 17th May 2021].

[17] M. A. Mazidi, J. G. Mazidi and R. D. McKinlay, THE 8051 MICRO CONTROLLER AND EMBEDDED SYSTEMS Using Assembly and C, 2nd ed., Pearson, 2006.

[18] M. A. Mazidi, the 8051 Microcontroller and Embedded System.

[19] "Theengineeringprojects," 4th October 2018. [Online]. Available: https://www.theengineeringprojects.com/2018/10/introduction -to-led-light-emitting-diode.html. [Accessed 27th August 2021].

[20] "LEDs Magazine," 1st September 2004. [Online]. Available: https://www.ledsmagazine.com/leds-ssl-design/materials/article/16701292/what-is-an-led. [Accessed 27th August 2021].

[21] "Electronicsforu," 4th June 2019. [Online]. Available:https://www.electronicsforu.com/resources/7-segment-display-pinout-understanding. [Accessed 27th August 2021].

[22] "Robu," 12th February 2021. [Online]. Available: https://robu.in/seven-7-segment-display-interfacing-arduino/. [Accessed 27th August 2021].

[23] M. A. Mazidi, the 8051 Microcontroller and Embedded System, 2008.